

■はじめに

本書は、2010年に初版発行された『プログラマのための文字コード技術入門』の改訂版です。初版は幸いにも多くのご支持をいただき、6回の増刷を重ね、発行された年の東京の大型書店で発表された年間ランキングのコンピュータ書部門で第4位に入りました。

この改訂版では、初版の骨格はそのままに、この9年間の変化を反映して古くなった記述を改め、規格や文字政策、プログラミング言語・API等の最新版に追従するよう全面的に加筆修正しています。

この前書きに目を通していらっしゃる方には、初版を読み終えていて、改訂版を読むべきかを思案中の方もいるでしょう。文字コードの適用、仕様、原理原則の3つに分けて考えると、この9年間で、適用をめぐるのはプログラミング言語など変更がありました。文字コードの仕様の基本的な部分には大きな修正はありません。原理原則はまったく変わりません。したがって、文字コードというものの概要を知ることが目的の場合は、初版の知識だけでも大体のところは問題ないでしょう。一方、新たに追加された仕様など、知識を最新版へとアップデートしたい方にはこの改訂版が役立つでしょう。

文字コードはしばしば論争の種となるテーマですが、本書は特定のコード系の使用を推奨したり否定したりするものではありません。代わりに、各コード系の特徴や適した用途等を、言語表記の記号としての文字を符号化する観点から根拠に基づいて説明しています。

本書初版に寄せられた感想で予想外に嬉しかったことは、「おもしろい」という感想が多かったことです。本書は技術書ですから第一義的には技術的に役に立つことが求められますし、実際、役に立ったという感想も少なからずいただきました。一方、おもしろいということには、物事への興味を引き出し、より深い理解へ導く力があると筆者は考えます。本書がおもしろいという評価をいただいているのが、文字の符号化という営みの本質的な側面を本書が彫り出し得たためであるならば、著者として望外の喜びです。この改訂版がより多くの方にお楽しみいただけることを期待します。

2018年12月 矢野 啓介

■——はじめに ※初版前書きより。

文字コードは、ソフトウェア技術者にとって必須の知識です。プログラミングをするうえでも、データフォーマットなどを設計するうえでも必要です。しかし、文字コードの知識を体系的に学ぶ機会というのは滅多にありません。断片的な手がかりを元に Web で検索するなどして調べるといった対応をすることが多いと思います。

本書は、ソフトウェア技術者をおもな対象として、文字コードの基礎知識をなるべく筋道立てて説明しようと試みたものです。前半の第1章から第4章までは、文字コードの基本的な概念から始めて、現在日本で使われているものを中心として各種の文字コードを紹介します。後半の第5章から第8章までは、いわば応用編として、コード変換や判別、インターネットでの扱い、プログラミング言語での扱い、そして典型的なトラブルについて説明します。

さまざまな文字コードを本書では取り上げますが、一概にどれを使うべきといった判断は本書は下しません。どの文字コードもそれなりの理由があって作られたものですから、自分のプロジェクトの置かれた状況に応じて有用なものを選んで使えば良いというのが本書のスタンスです。本書ではそれぞれの文字コードがどのような特徴を持っているか、どのような用途に向くかを説明します。また、本書では説明の根拠となる規格等への参照をできるだけ本文中に記すよう努めました。より詳しく知りたい方は、そうした規格等に直接あたって調べることができます。

本書の内容について、内田明氏、高橋征義氏、武者晶紀氏より有益なご助言をいただきました。深く感謝いたします。もちろん、本書の内容に誤りがあれば著者の責任であることはいうまでもありません。

本書の記述のうち意見にかかわるものは、筆者個人の見解であり、いかなる組織を代表するものでもないことをお断りしておきます。

本書が読者の文字コード理解の一助になれば幸いです。

2010年1月 矢野 啓介

■—— 第2版改訂における、おもな変更点

第2版改訂に伴い、おもに以下の内容の追加・更新を行いました。

- 2010年の改正常用漢字表に対応
- JIS X 0208とJIS X 0213の2012年改正に対応
- ISO/IEC 10646のUCS-4の変更に対応
- Unicodeに追加された変体仮名の説明
- Unicode 11に基づいた説明
- 本書初版では標準化作業中だったUnicode絵文字を、最新版に基づいた説明に変更
- Webブラウザの動作例を2018年時点で使用されているバージョンに更新
- 紹介するコマンドラインツールを最近よく使われるものに見直し
- HTML5の仕様を反映
- YAMLとJSONについて記述
- Javaの対象バージョンを初版の6から、2018年時点で使用の多い8に変更（一部、9にも対応。ただし、それ以降の版でも通用するでしょう）
- Ruby 1.9以降の説明を2.5に基づいて更新
- Javaのデフォルト文字コードの指定方法の説明を追加
- 文字化けのよくあるパターンを紹介
- Unicodeの標準化異体シーケンス(*Standardized Variation Sequence*)の説明を追加
- WebサイトやメールにおいてUTF-8の使用が増えたことを反映した修正

■—— 謝辞

今回の改訂にあたり、初版に引き続き、内田明氏、高橋征義氏、武者晶紀氏より再度有益なご助言をいただきました。心より感謝いたします。

はじめに.....	iii
はじめに(初版前書きより).....	iv
第2版改訂における、おもな変更点.....	v

第1章

文字とコンピュータ..... 001

1.1	コンピュータで文字を扱う基本	003
	文字コードとフォント.....	003
	図形を交換するのではなく、符号を交換する.....	003
	文字の形の細部は伝わらない.....	004
1.2	文字を符号化すること	005
	コンピュータで情報を扱う基礎.....	005
	文字を符号化する例.....	006
1.3	文字集合と符号化文字集合	007
	何文字必要か.....	007
	文字の集合.....	008
	文字の集合に符号を振る.....	009
	符号化文字集合とは.....	009
	実用的な符号化文字集合の例.....	010
	一意な符号化 文字コードの原則.....	012
	符号化文字集合を実装するとは.....	013
	文字化け.....	014
	外部コードと内部コード.....	015
	規格における定義 符号化文字集合、符号.....	016
1.4	制御文字 文字ではない文字	017
	文字コードにあるのは文字だけではない.....	017
	おもな制御文字.....	017
1.5	文字コードはなぜ複雑になるのか	018
	文字コードを複雑化させる二つの理由.....	018
	過去の経緯の積み重ね.....	018
	文字そのものの難しさ.....	019
	文字コードの複雑さを理解するために.....	020
1.6	まとめ	020

	第2章	
	文字コードの変遷	021
2.1	最もシンプルな文字コード ASCII、ISO/IEC 646	023
	7ビットの1バイトコードで文字を表すASCII	023
	ASCIIの各国用の変種 各国語版ISO/IEC 646	023
	ISO/IEC 646とJIS X 0201	025
2.2	文字コードの構造と拡張方法を定める ISO/IEC 2022	025
	ISO/IEC 2022の登場 8ビットコード、2バイトコード	026
	ASCIIを拡張する	026
	8ビットの使用 ISO/IEC 2022の枠組み、CL/GL、CR/GR	027
	符号化文字集合の呼び出しの概念	028
	複数バイト文字集合	029
	符号化文字集合の組み合わせ・切り替え	030
	ISO/IEC 2022とエスケープシーケンス	030
	2022≠エスケープシーケンスによる切り替え	031
	ISO/IEC 2022と符号化方式	032
2.3	2バイト符号化文字集合の実用化 JIS X 0208、各種符号化方式	033
	JIS X 0208 漢字を扱う	033
	各種「符号化方式」の成立	034
	1バイトコードに2バイトコードを組み合わせたい	034
	Shift_JISやEUC-JP、ISO-2022-JPの登場	035
	東アジアでの普及	035
2.4	1バイト符号化文字集合の広がり ISO/IEC 8859、Latin-1	036
	ヨーロッパ各地域向けの文字コード	036
	ISO/IEC 8859、ISO/IEC 8859-1、Latin-1	036
	1バイト文字集合の乱立	037
2.5	国際符号化文字集合の模索と成立 Unicode、ISO/IEC 10646	038
	世界中の文字を一つの表に収める	038
	ISO/IEC 10646とUnicodeの誕生と統合	039
	Unicodeの拡張と各種符号化方式の成立 UTF-16、UTF-8	040
	国際符号化文字集合の現状	041
	Unicodeの使用状況 OSの内部コードやWebページと、その他の状況	042
2.6	まとめ	042
	Column 字形と字体	043
	Column 常用漢字表の改正と文字コード	044

第3章 代表的な符号化文字集合

3.1	ASCIIとISO/IEC 646 最も基本的な1バイト文字集合	047
	ASCIIとISO/IEC 646国際基準版	047
	各国版のISO/IEC 646	048
3.2	JIS X 0201 ラテン文字と片仮名の1バイト文字集合	049
	JIS X 0201の概要	049
	ラテン文字集合	049
	JIS X 0201の片仮名集合、濁点・半濁点	049
	ASCIIとの違い 円記号とバックスラッシュ、オーバーラインとチルダ	051
3.3	JIS X 0208 日本の最も基本的な2バイト文字集合	052
	JIS X 0208の概要 ISO/IEC 2022準拠	052
	符号の構造 2バイトのビット組み合わせ	053
	文字集合の特徴	054
	記号類	055
	ギリシャ文字	055
	キリル文字	055
	ラテン文字	056
	平仮名・片仮名	056
	漢字 第1水準・第2水準	057
	過去の改正の概略	058
	1983年改正	058
	字体の変更(簡略化)と符号位置の入れ替え	059
	1990年改正	060
	1997年改正 包摂規準	061
	包摂規準の明示	061
	JIS X 0208(97JIS)の包摂規準の活用	062
	JIS X 0208(97JIS)の包摂規準の生い立ち	063
	漢字の包摂規準を理解する	064
	漢字の典拠調査 幽霊漢字の退治	065
	JIS X 0208:1997の符号化方式	065
	外字・機種依存文字の問題	066
3.4	JIS X 0212 補助漢字	068
	JIS X 0212の概要	068
	文字集合の特徴	069
	非漢字	069
	Column 「Unicodeで(他の符号化文字集合を)実装」という表現の問題	070
	漢字	070
	JIS X 0212と符号化方式 Shift_JISで扱えない	071
3.5	JIS X 0213 漢字第3・第4水準への拡張	072

	JIS X 0213の概要.....	072
	漢字集合1面、漢字集合2面.....	073
	文字集合の特徴.....	074
	一般の印刷物でよく使われる記号類.....	075
	13区の機種依存文字と互換の文字.....	076
	ラテン文字・発音記号.....	077
	日本語のローマ字表記に必要な文字.....	077
	発音記号として使われる文字.....	077
	その他のダイアクリティカルマーク付きの文字など.....	078
	合成用のダイアクリティカルマーク.....	079
	ASCIIとの互換性のための文字.....	079
	アイヌ語表記用片仮名.....	080
	鼻濁音表記用の平仮名・片仮名など.....	082
	漢字(第3・第4水準).....	083
	地名や人名、学校教科書に使われる漢字.....	083
	収録された文字の収集にあたって.....	084
	峯問題の字体の新規追加による解決.....	085
	199の包摂規準.....	085
	人名用漢字のすべて JIS X 0208で包摂されていた微小な差を分離したもの.....	085
	1983年改正で字体が大きく変更された漢字29文字の変更前の字体.....	086
	漢字のへんやつくりなどの字体記述要素.....	087
	符号化方式.....	087
	符号化方式をめぐる論議 規定か、参考か.....	088
	2004年改正の影響 表外漢字字体表と例示字形.....	089
	Unicodeとの対応関係 表外漢字UCS互換.....	089
	ソフトウェアのJIS X 0213対応状況.....	091
3.6	ISO/IEC 8859シリーズ 欧米で広く使われる1バイト符号化文字集合	092
	ISO/IEC 8859(シリーズ)の概要.....	092
	Latin-1 ISO/IEC 8859-1.....	093
	ノーブレークスペース(NBSP)とソフトハイフン(SHY).....	094
	Latin-2 ISO/IEC 8859-2.....	095
	その他のパート.....	096
3.7	UnicodeとISO/IEC 10646 国際符号化文字集合	097
	UnicodeおよびISO/IEC 10646(UCS)の概要.....	097
	符号の構造 UCS-4、UCS-2、BMP.....	098
	Unicodeの符号位置の表し方.....	099
	基本多言語面(BMP).....	100
	その他の面.....	102
	面01 SMP.....	102
	変体仮名.....	102
	面02 SIP.....	103
	面0E.....	103
	結合文字 1文字が1符号位置ではない.....	104
	既存の符号化文字集合との関係.....	105
	Unicodeにおける文字名の定義 各文字に一意な名前を与える.....	105

Column	ちょっと気になるUnicodeの文字名	105
	ISO/IEC 8859-1との関係	106
	全角・半角形	106
	漢字統合 CJK統合漢字	108
	原規格分離規則	108
	統合漢字の数	109
	漢字統合と適切なフォントの選択	110
	互換漢字	110
	互換漢字の領域	111
	互換漢字と正規化	112
	JIS X 0213との関係 プログラムで処理する上での注意点	112
	①BMP以外の面の漢字の存在	113
	②結合文字の使用の必要	114
	③互換漢字の正規化の問題	116
	絵文字	117
	絵文字とは	117
	符号位置概要	117
	複数符号位置による装飾	118
	国旗の特殊な符号化	120
	絵文字の形の違い	121
	絵文字に未来はあるか	122
Column	UnicodeとUTF-8とUCS-2の関係	124

第4章 代表的な文字符号化方式

4.1	JIS X 0201の符号化方式	127
	JIS X 0201の符号化方式の使い方	127
	8ビット符号	127
	7ビット符号	128
4.2	JIS X 0208の符号化方式	130
	JIS X 0208で定められた符号化方式	130
	漢字用7ビット符号	131
	符号の構造	131
	漢字用7ビット符号の特徴	132
	適した用途	133
	EUC-JP	133
	符号の構造	133
	国際基準版・漢字用8ビット符号との関係	135
	EUC-JPの特徴と注意	136
	重複符号化の問題	136
	適した用途	137
	ISO-2022-JP	137

符号の構造.....	138
符号の性質.....	140
適した用途.....	140
Shift_JIS.....	140
符号の構造.....	141
Shift_JISの計算方式.....	142
Shift_JISの問題点.....	143
重複符号化の問題.....	143
適した用途.....	144
機種依存文字付きの変種.....	144
4.3 Unicodeの符号化方式.....	145
UTF概説.....	145
UTF-16.....	146
符号の構造.....	146
サロゲートペア.....	147
UTF-16の計算方法.....	147
UCS-2との関係.....	148
UTF-16のバイト順の問題 ビッグエンディアンとリトルエンディアン.....	148
BOM (バイト順マーク).....	149
適した用途.....	150
UTF-32.....	150
符号の構造.....	150
UCS-4との関係.....	151
UTF-32の特徴.....	151
適した用途.....	152
UTF-8.....	152
符号の構造.....	152
計算方法.....	152
ASCIIとの互換性 UTF-8の特徴.....	153
冗長性の問題.....	154
BOM付きUTF-8の問題.....	154
CESU-8とModified UTF-8.....	155
適した用途.....	155
<u>Column</u> 機種依存文字における重複符号化.....	156

第5章 文字コードの変換と判別..... 157

5.1 コード変換とは.....	159
なぜ変換が必要か.....	159
変換のツール.....	159
iconv.....	160
変換できない場合.....	161

	nkf.....	162
	変換の原則	162
	異なる文字集合体系の間の変換の問題.....	163
	コード変換と文字変換	166
5.2	変換の実際 変換における考え方	167
	コード変換の処理方法	168
	アルゴリズム的な変換	168
	JIS X 0208の符号化方式の変換.....	168
	ISO-2022-JPとEUC-JPの間の変換 エスケープシーケンスと0x80の足し引き.....	168
	Shift_JISの関係する変換 区点番号を介した計算.....	170
	JIS X 0201とASCIIの違いの問題 Shift_JISの0x5C、0x7E.....	170
	文字コードの定義に忠実なコード変換とその問題.....	171
	Unicodeの符号化方式の変換.....	172
	UTF-8からの変換.....	172
	UTF-16からの変換.....	173
	テーブルによる変換	174
	JIS X 0208とUnicodeの間の変換.....	174
	JIS X 0208とASCII/JIS X 0201の間の変換.....	176
	JIS X 0201ラテン文字集合の変換の例題.....	176
	ハイフンマイナスの問題.....	177
	JIS X 0201片仮名集合の場合.....	178
	変換の必要性 使い勝手の向上のために.....	178
5.3	文字コードの自動判別	179
	自動判別の例	179
	判別のツール nkf.....	180
	なぜ自動判別できるか	180
	BOMによる判別.....	180
	エスケープシーケンスによる判別.....	181
	バイト列の特徴を読む EUC-JPとShift_JISの判別例.....	182
	自動判別を助けるテクニック.....	183
	自動判別の限界	185
5.4	まとめ	186

第6章

インターネットと文字コード..... 187

6.1	電子メールと文字コード	189
	メールの基本はASCII 日本語は7ビットのISO-2022-JPで.....	189
	MIME	190
	メールを多言語に拡張する.....	190
	charsetパラメータで文字コードを指定する.....	191
	charsetパラメータの値.....	192

誤ったcharset指定.....	193
<code>Column</code> character setという用語.....	194
テキストをさらに符号化する.....	195
Content-Transfer-Encodingフィールド.....	195
quoted-printable.....	196
base64.....	197
base64による符号化のしくみ.....	197
ヘッダの符号化 B符号化とQ符号化.....	199
nkfによる復号.....	200
添付ファイル名の符号化.....	201
添付ファイル名のトラブルの原因.....	201
添付ファイル名の文字化けへの対処法.....	203
日本語メールの符号化の現在.....	203
6.2 Webと文字コード.....	204
HTML.....	204
HTMLで用いる文字.....	204
SGMLとしての背景.....	205
HTMLの文字参照.....	206
文字コードの指定方法 head要素の中のmeta要素.....	207
lang属性の影響.....	208
統合漢字を描画し分ける.....	208
言語情報は書体選択の役に立つか.....	210
CSS.....	211
文字コードの指定方法.....	211
Unicode文字の参照.....	212
XML.....	212
XMLで用いる文字.....	213
XMLの文字参照.....	214
文字コードの指定方法.....	214
XML宣言.....	214
XHTMLの場合.....	215
YAMLとJSON.....	215
URL.....	216
URL符号化.....	216
HTML/XMLの中のURL.....	217
HTTP.....	218
HTML文書内部の文字コード指定が抱える問題点.....	218
HTTPヘッダによる文字コードの指定.....	219
Webサーバにおける設定.....	220
HTTPヘッダの確認方法.....	221
HTML フォーム (CGI).....	222
フォームから入力されるテキストの文字コード.....	222
送信用の文字コードで符号化できない文字の扱い.....	224
6.3 まとめ.....	225

第7章 プログラミング言語と文字コード 227

7.1 Java 内部処理をUnicodeで行う 229

Javaにおける文字はすべてUnicode 229

Javaの文字列と文字 229

StringクラスとCharacterクラスとchar型 229

ソースコードの中の文字 230

コンパイル時のコード変換 230

Unicodeエスケープ 231

JavaはUnicodeを知っている 232

文字の属性を調べる 233

大文字・小文字 Character.isLowerCase(char)メソッド、他 233

数字・文字 Character.isDigit(char)メソッド 234

Unicodeブロック Character.UnicodeBlockクラス 235

サロゲートペアにまつわる問題 char単位で文字を扱うメソッド 236

サロゲートペアへの対応 charからintへ 238

入出力における文字コード変換 240

Reader/Writerクラスによる変換 240

文字コードを指定した入出力 InputStreamReader/OutputStreamWriterクラス 241

Javaで扱える文字コード 241

プラットフォームのデフォルトの文字コードを得る 243

デフォルトの文字コードを指定する 243

プロパティファイルの文字コード 244

native2ascii 244

プロパティエディタ プロパティファイル編集用のツール 245

XML形式のプロパティファイル 246

Propertiesクラス 246

リソースファイル プロパティファイルを国際化のために用いる 246

JSPと文字コード 247

pageディレクティブによる指定 247

Windowsの場合の問題 MS932変換表とSJIS変換表 248

よく問題になる例 ~ (波ダッシュ) 249

3つの対処法 入力/出力におけるUnicode変換の食い違いを解消する 251

①入力時の変換を、SJIS変換表に揃える 251

②MS932変換表を使う 252

③Javaプログラムで置換したうえでSJIS変換表で出力する 253

文字コード変換器の自作方法 253

ソートの問題 テキスト処理① 254

文字コードによるソート順 255

文字コード順以外によるソートの必要性 言語や国・地域を考慮する 257

Collatorクラスの使用 257

CollationKeyによる性能改善 259

自然な区切り位置の検出 テキスト処理② 259

何が問題か Javaのcharと、結合文字やサロゲートペア 260

BreakIteratorクラス 適切な区切り位置を検出する 261

7.2	Ruby 1.8 シンプルな日本語化	262
	バージョン1.8までのRubyは、ASCIIが基本	262
	Ruby 1.8の文字列	263
	文字列の長さ	263
	バイト列としての操作	263
	文字列の操作	264
	文字列の比較とソート	264
	jcodeによる複数バイト文字対応	266
	文字コードの指定	266
	指定方法 -kオプション、\$KCODE	267
	正規表現のマッチング 文字コードの指定を適切に行う	268
	\$KCODEによる違い	268
	正規表現ごとの文字コード指定	269
	文字列を文字単位に切り分けるイディオム	270
	文字コードの指定を間違えたと何が起るか	270
	JIS X 0213を使う	272
	コード変換ライブラリ	273
	NKF	273
	コード判別	273
	Kconvクラス	274
	Iconvクラス	275
7.3	Ruby 1.9以降 CSI方式で多様な文字コードを処理	276
	拡張されたRuby 1.9の文字関連処理	276
	スクリプトの文字コードの指定 マジックコメント	276
	Ruby 1.9の文字列	277
	自分の符号化方式を知っている	277
	文字列の連結	278
	Unicodeエスケープ	279
	文字単位の操作	279
	文字列の長さ	281
	Unicodeの結合文字やサロゲートの扱い	281
	結合文字を含めた「1文字」をとる	282
	入出力の符号化方式 IOクラス	283
	入出力における文字コードの指定	283
	Encodingクラス	285
	Ruby 1.9のコード変換	285
	String#encodeメソッド	286
	挙動の制御	286
	変換できない文字の扱い 挙動の制御①	286
	XMLのメタ文字のエスケープ 挙動の制御②	287
	Encoding::Converterクラス	287
7.4	まとめ	288

第8章

はまりやすい落とし穴とその対処

8.1	トラブル調査の必須工具 16進ダンプツール	291
	データのバイト値を検査する	291
	od 16進ダンプのツール	291
	その他のツール hd, xxd	292
8.2	文字化け	292
	文字化けのよくあるパターン	292
	ラベルと本体の不一致による文字化け	294
	機種依存文字に起因する文字化け	294
	文字化け防止の原則	295
8.3	改行コード	297
	改行コードに起因するトラブル	297
	1つのファイル中の混在	297
	想定外の改行コードの使用	298
	改行コードの変換	298
	nkfコマンドによる改行コードの変換	299
	trコマンドによる対応	299
8.4	「全角・半角」問題	300
	「全角・半角」で何が問題になるのか	300
	問題の本質	301
	区別のはじまり かつての機器のテキスト表示の制約条件	301
	用語の本来の意味 印刷用語の全角・半角	302
	文字コードは「全角・半角」を決めていない 1バイトの「A」、2バイトの「A」	302
	「(いわゆる)全角・半角」の存在は便利なのか	303
	「全角・半角」問題への対応 利用者に「全角・半角」を意識させない	304
	求められる文字入力プログラム 文字コードにおける一意な符号化という原則	305
	入力文字の検証 アプリケーション側の対処法①	306
	重複符号化された文字の同一視 アプリケーション側の対処法②	306
8.5	円記号問題	307
	円記号問題とは何か	307
	ASCIIとJIS X 0201の違い	308
	円記号問題の顕在化	308
	Webブラウザ上の表示	309
	Unicodeとの変換による問題 単なる表示上の問題では済まなくなる	310
	対処のための注意点	312
	EUC-JPの場合	312
	文字入力の際の注意	313
	チルダとオーバーラインについての注意	313
	円記号問題は解決できるか	314

	問題の本質 0x5Cの意味の違いを厳密に運用する.....	314
	解決のための思考実験.....	314
8.6	波ダッシュ問題	316
	波ダッシュ問題とは何か.....	316
	現象の例.....	317
	波ダッシュとは.....	317
	チルダとは.....	318
	問題の原因 WAVE DASHとFULLWIDTH TILDE.....	319
	変換の妥当性を検証する JISの1区33点とU+301Cの対応付け.....	320
	Unicodeの例示字形.....	320
	FULLWIDTH TILDEの存在.....	321
	Windowsの実装.....	321
	三つの対処案.....	322
	①Unicodeに変換しない.....	322
	②コード変換を揃える.....	322
	③Unicode間で変換する.....	323
	波ダッシュ以外の文字 変換による問題が発生しがちな文字.....	324
8.7	まとめ	325
	Appendix	327
A.1	ISO/IEC 2022のもう少しだけ詳しい説明	328
	符号化文字集合のバツファ.....	328
	指示と呼び出し.....	328
	94文字集合と96文字集合.....	329
	エスケープシーケンス.....	330
	符号化方式の実際.....	330
	EUC-JP.....	331
	ISO-2022-JP.....	331
A.2	JIS X 0213の符号化方式	333
	既存の資産を活かしつつJIS X 0213の利点を享受するために.....	333
	漢字用8ビット符号.....	334
	適した用途.....	335
	EUC-JIS-2004.....	335
	「国際基準版・漢字用8ビット符号」との関係.....	337
	適した用途.....	337
	ISO-2022-JP-2004.....	338
	包摂規準の変更による旧規格使用の制限.....	339
	適した用途.....	339
	Shift_JIS-2004.....	340
	適した用途.....	341
A.3	諸外国・地域の文字コード概説	342

中国	GB 2312とGB 18030	342
	GB 2312	342
	GB 18030	343
韓国	KS X 1001	343
北朝鮮	KPS 9566	344
台湾	Big5とCNS 11643	345
	Big5	345
	CNS 11643	346
香港	HKSCS	347
ロシア	KOI8-R	347
A.4	Unicodeの諸問題	349
	正規化 いつのまにか別の文字に変わる?	349
	問題	349
	正規化	350
	NFKC、NFKD	351
	正規化によって別の文字に移される文字	352
	日本語環境への影響	353
	互換漢字の扱い	354
	Javaにおける正規化	354
	ファイル交換の際のトラブル	355
	器問題 統合漢字と互換漢字の複雑な関係	355
	拡張B 日本風、台湾風の器器	356
	Webブラウザの表示例	357
	異体字セレクタ 「正しい字体」への欲求	358
	文字コードは文字の形を抽象化する	358
	異体字を指定する	359
	IVS	359
	互換漢字の代替手段としての異体字セレクタ	360
	プログラム上の対処	361
	書字方向の制御によるファイル名の偽装	361
	右から左に書く文字	362
	ファイル名の偽装	362
	偽装のしくみ	362
	Windowsによる実験	363
	Windowsにおける対策	364
A.5	Unicodeの文字データベース UnicodeData.txtとUniHan Database	366
	UnicodeData.txt	366
	文字の種別の判別	366
	UniHan Database	367
A.6	規格の入手・閲覧方法ならびに参考文献	368
	参考文献	370
	索引	372